The ARM® Mali™ -T880 Mobile GPU

Ian Bratt
ARM Media Processing Group

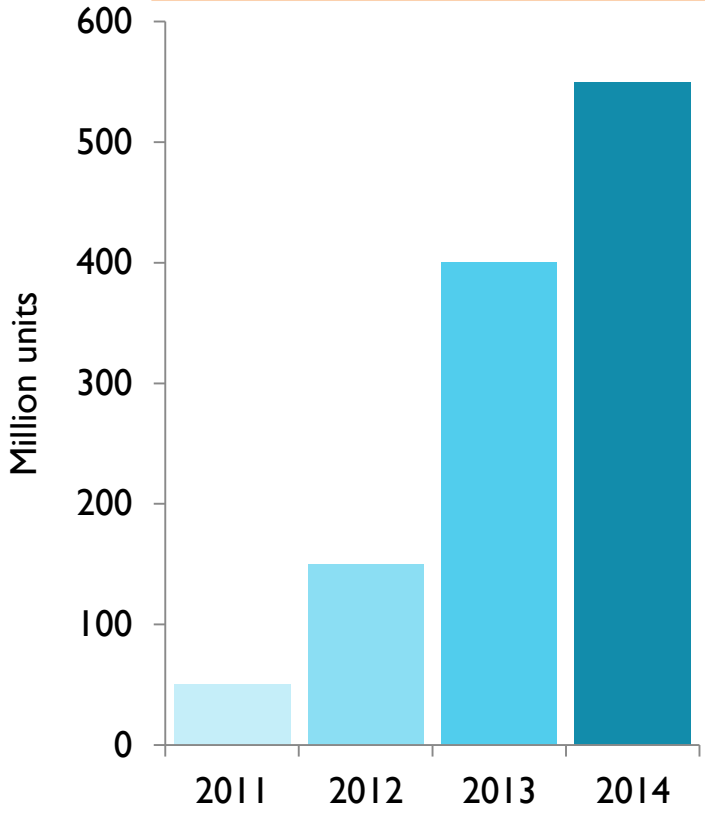
ARM® Mali™ Success

73 licensees

Total
114
licenses

25
new Mali
licenses
in FY14

550M
Mali–based GPUs
shipped in 2014



Mali is in
75%
of DTVs…

…
>50%
of Android
tablets…

…
>35%
of
smartphones

ARM®

ARM Mali-T880 Outline

- Overview
- Rendering flow
- Scalability
- Block diagrams
  - GPU
  - Shader core
- Bandwidth saving mechanisms
  - Transaction Elimination
  - ASTC
  - AFBC
- Summary

Mali-T880 Overview

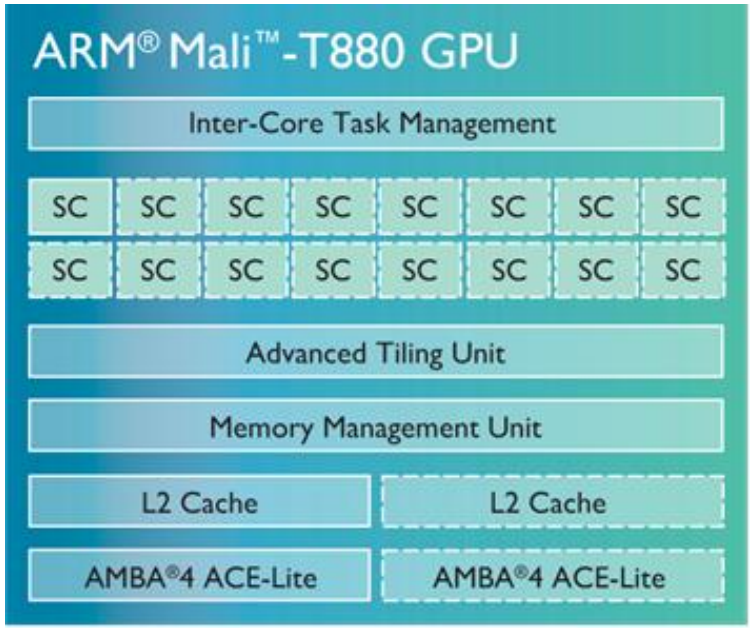
- **Targeted towards mobile phones, tablets, set top boxes, etc…**

- **Evolution of Mali 'Midgard' Tri-pipe architecture**
  - Unified shader architecture
  - Unified memory architecture
  - Deferred rasterization
  - Improved energy efficiency and performance over Mali-T760

- **Multi-processor scalable graphics performance**
  - Scalable to 16 coherent shader cores
  - Mali-T860 - 2 ALUs per shader core
  - Mali-T880 - 3 ALUs per shader core, for +50% compute performance

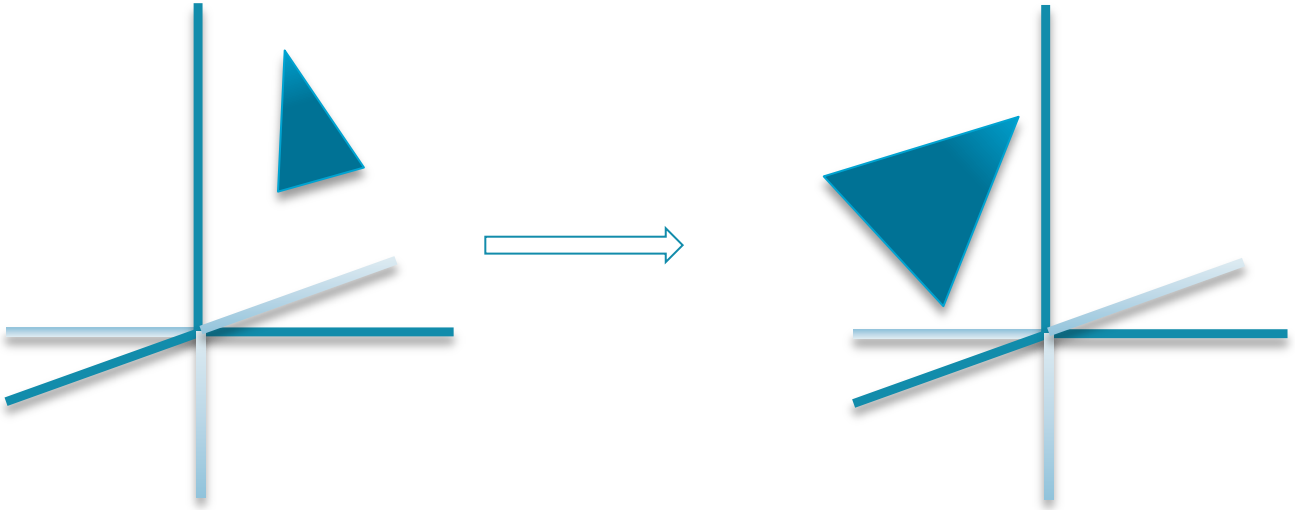
- **Improved integration with Video and Display IP**
  - 8/10bit YUV input & output
  - Video texture crop

- **Support for Major APIs**
  - Khronos OpenGL ® ES 3.1/3.0/2.0/1.1
  - Khronos OpenCL™ 1.2/1.1
  - Microsoft Windows DirectX ®11
  - Vulkan™ (support planned)



Mali-T880 Rendering Flow - Background

- Geometry processing (simplified)
  - Transform 3D geometry from one coordinate system to another
  - Perform additional calculations
    - Normals, etc.
  - Project onto 2D render target
  - In Mali vocab, we call these operations "Vertex Jobs"



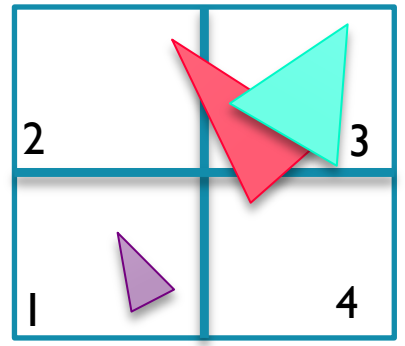
- Fragment processing (simplified)
  - For each pixel covered by a primitive, generate the final color for the render target
    - Rasterization and fragment shading
    - Be sure to maintain primitive order
  - In Mali vocab, we call these operations "Fragment Jobs"

ARM®

Mali-T880 Rendering Flow - Tiling

- Between Geometry and Fragment processing, Mali GPUs perform a phase called "Tiling"
  - Only begins once all geometry processing is complete
  - Tiling divides the final render target into small regions called "tiles"
  - The Tiling phase determines which set of primitives may contribute to pixels within a given tile
    - Note that primitive ordering must be preserved
  - The output of the tiling is a list, for each tile, containing the primitives which may contribute to said tile

Render Target

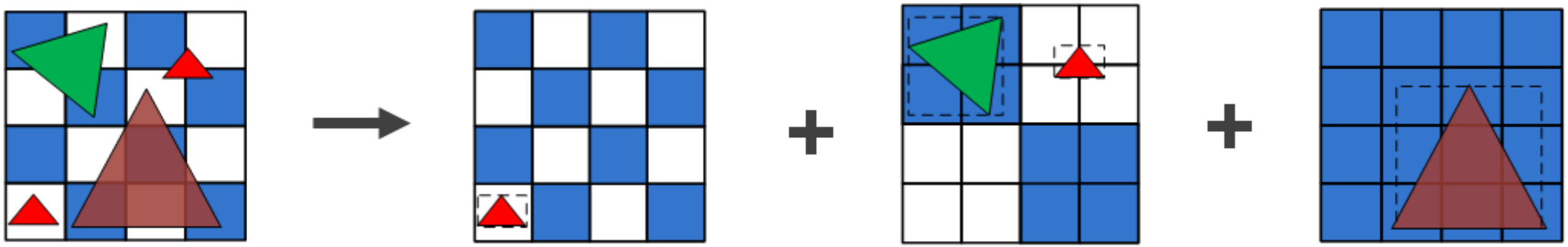


- Geometry Processing
- Tiling
- Fragment Processing

ARM®

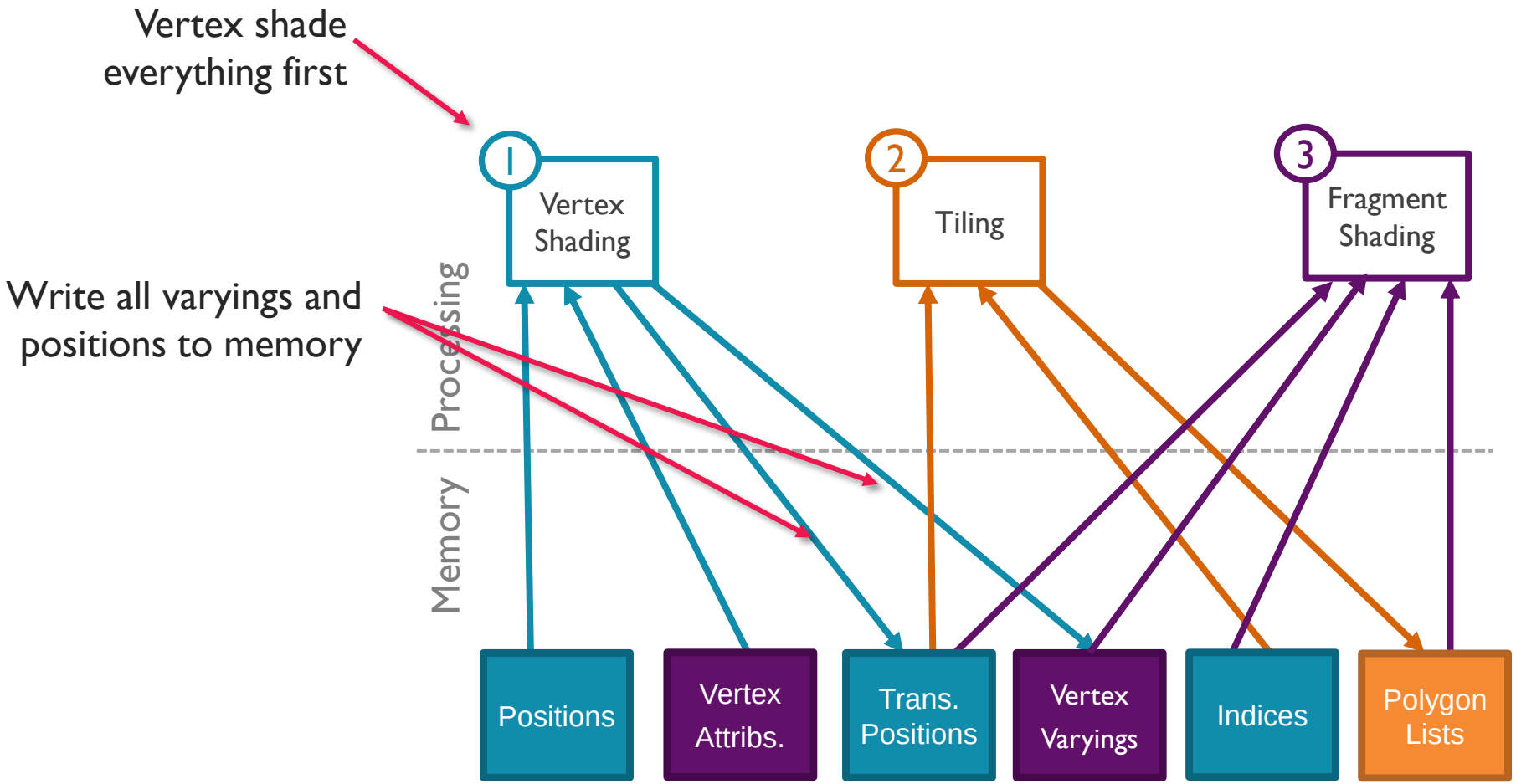
Mali-T880 Rendering Flow - Hierarchical Tiling

- Using only a single list per tile is wasteful. Large primitives would have to go into multiple lists
- Midgard implements hierarchical tiling for mapping primitives to screen coverage
- Multiple bin sizes efficiently decompose primitives for shading:



- Primitive size and screen resolution no longer affect bin count
  - Reduces memory bandwidth
- Makes memory footprint more predictable:
  - Now dependent on scene complexity, not primitive size
- Fragment shading is done at a tile granularity, and requires traversing all primitive lists which cover the tile

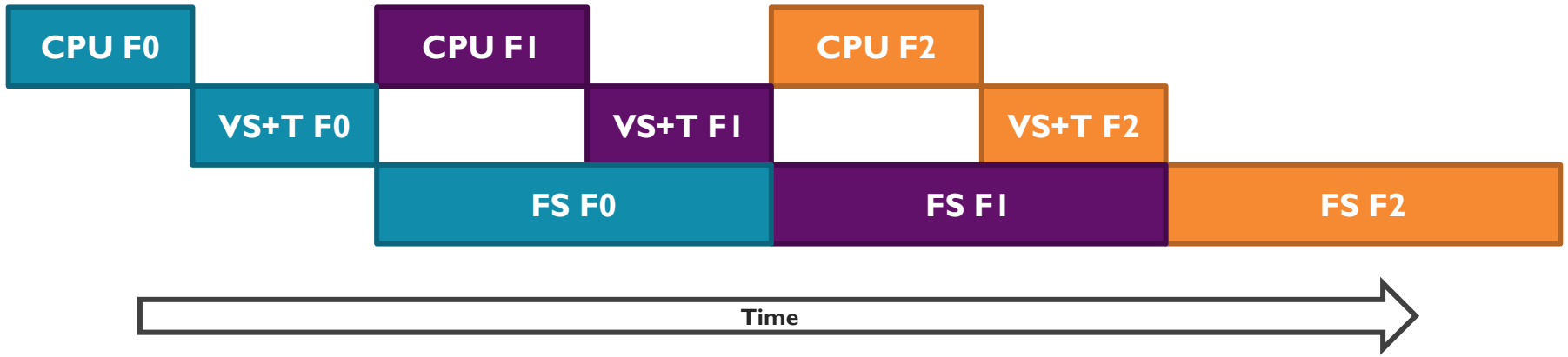
Mali-T880 Rendering Flow - Putting it all together



ARM®

Mali-T880 Rendering Flow - Pipelining Vertex and Fragment Jobs

- Mali GPU functionality is configured using descriptors
  - Memory-resident data structures
  - Control most aspects of GPU functionality
  - Very little is controlled via registers
- Mali GPUs support simultaneous vertex and fragment shading
- Vertex and Tiling jobs are sent to the GPU as a single job
- Rendering is pipelined
  - Vertex shading for RT N+1 running at the same time as fragment shading for RT N



ARM®

Mali-T880 GPU Scalability

- Mali GPUs are used in numerous different markets
- Different markets have different performance requirements

- The Mali architecture is scalable, and built from the ground up to serve multiple different markets
- Accomplished via a 1 pixel/cycle building block (the shader core), and a scalable GPU architecture
- Customers may spend more area/power, in order to meet more demanding performance requirements

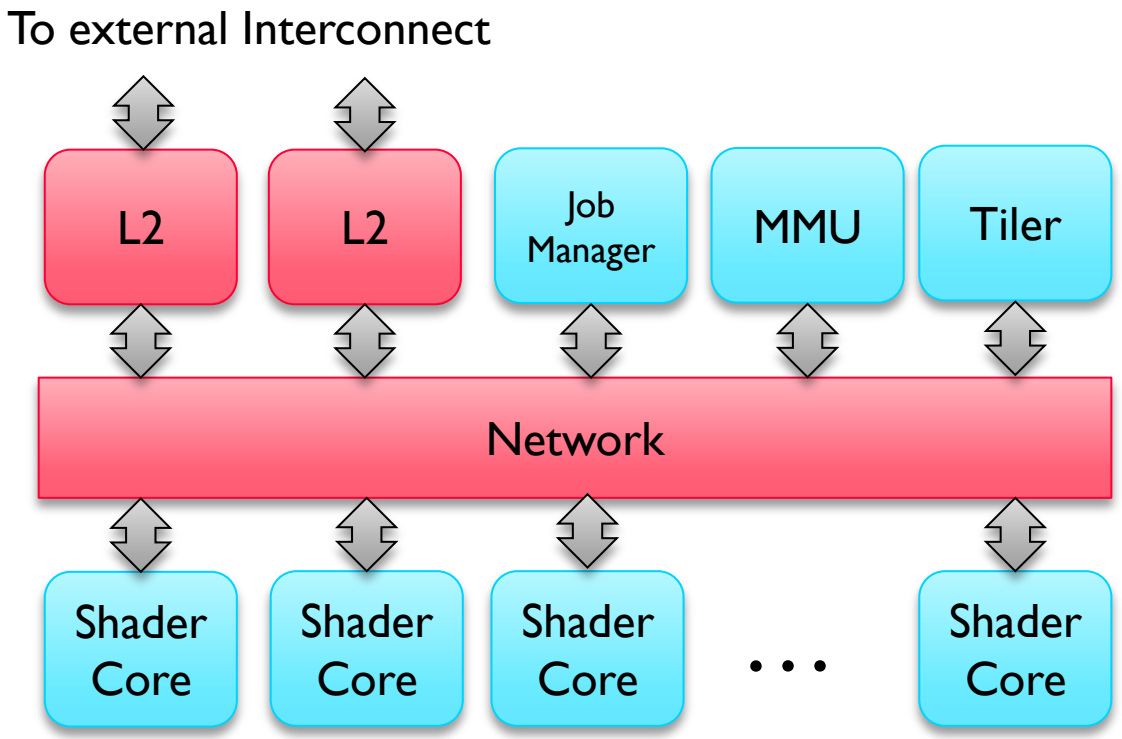
Performance

Area/Power

ARM®

Mali-T880 GPU Block Diagram - On-chip network

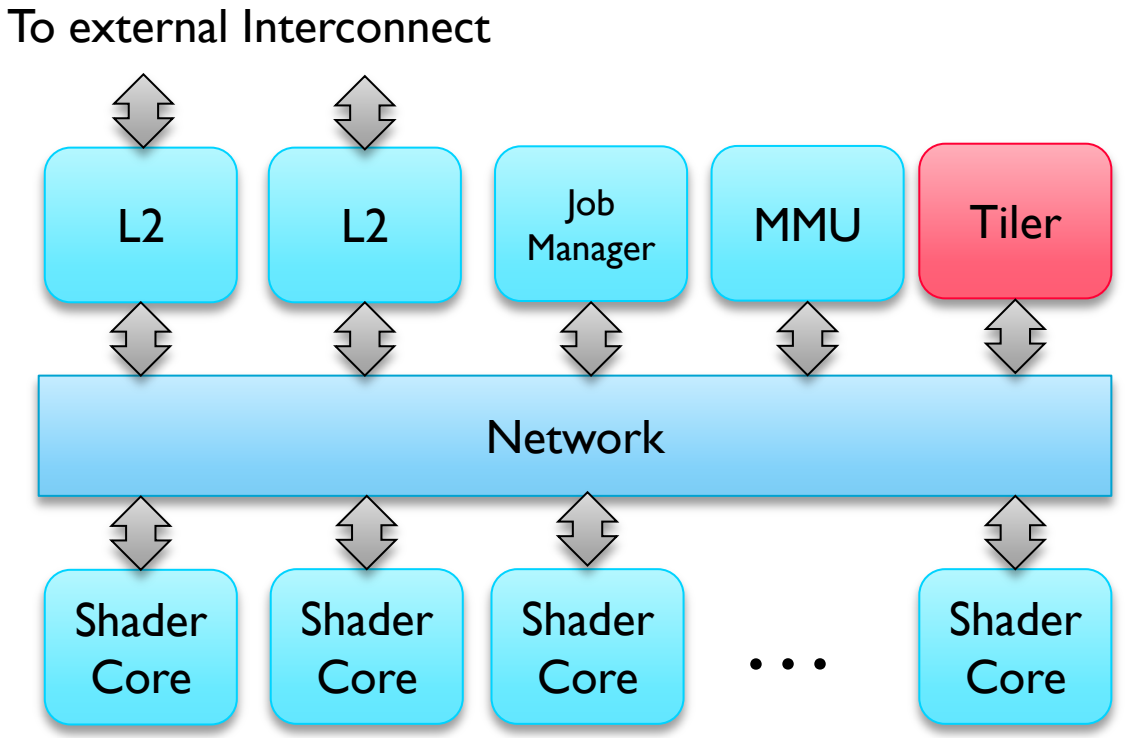
- Scalable on-chip network connects all components via architected protocol
  - Adding more cores adds more network BW
  - Network carries all memory accesses, as well as Job Manager messages and MMU translation requests
- Logical L2 implemented via multiple physical L2 slices with address interleaving.
  - Max of two L2 slices for MP16 configurations
  - Max size of 512KB per slice
  - Each L2 slice has a 128b AXI4/ACE-lite interface to the external interconnect
  - Supports IO-coherency



ARM®

Mali-T880 GPU Block Diagram - Tiler

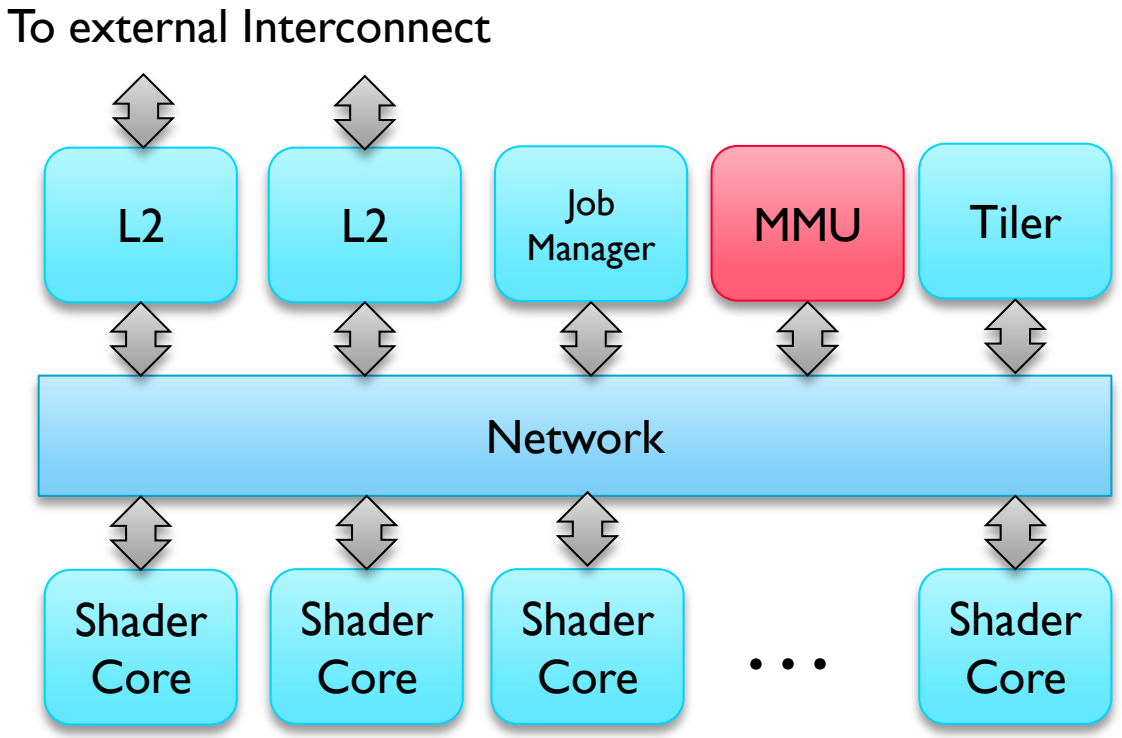
- Hierarchical Tiler
  - Fixed function block
  - Responsible for consuming the output of vertex shading and creating the polygon lists
  - Architectural performance of 1 triangle per cycle
  - Some results consumed by the Tiler may still be present within the shader cores, in which case, hardware coherency is used to properly snoop the data out of the shader cores …. (more on this later).



ARM®

Mali-T880 GPU Block Diagram - MMU

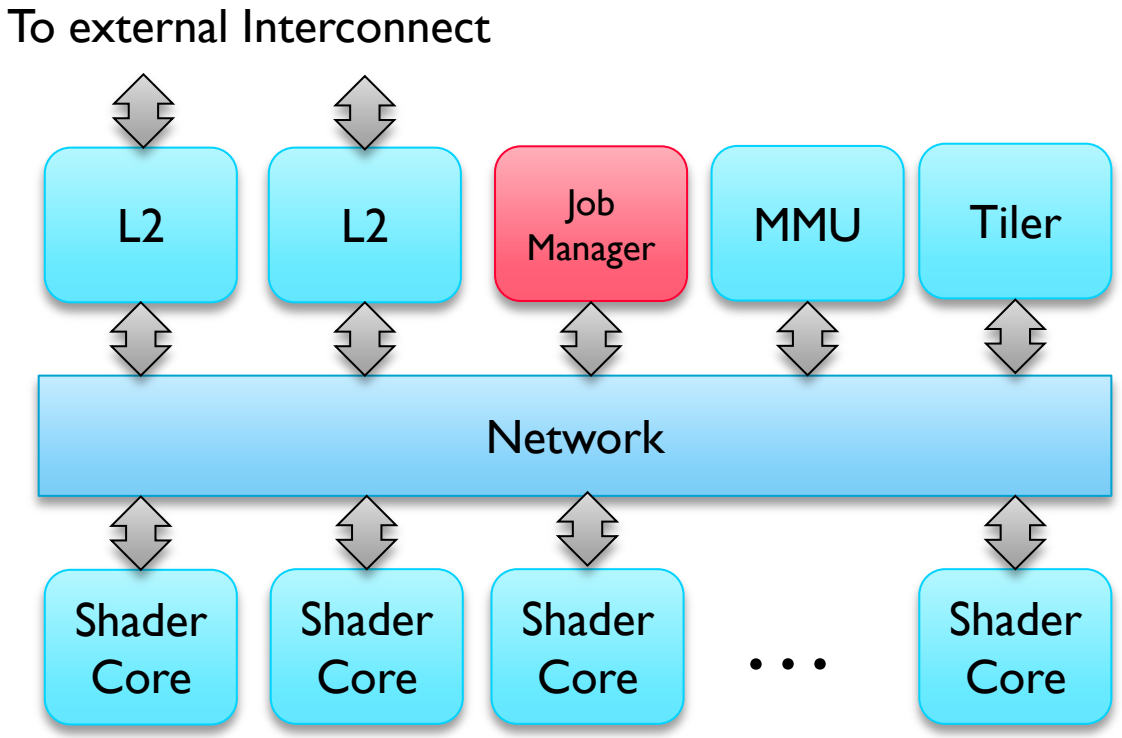
- Responsible for page table walks and address translation
  - Multiple uTLBs spread throughout the system
  - Two uTLBs per shader core
  - uTLB misses are sent to the MMU for servicing.
  - MMU performs the page table walk and responds to the uTLB with the translation



ARM®

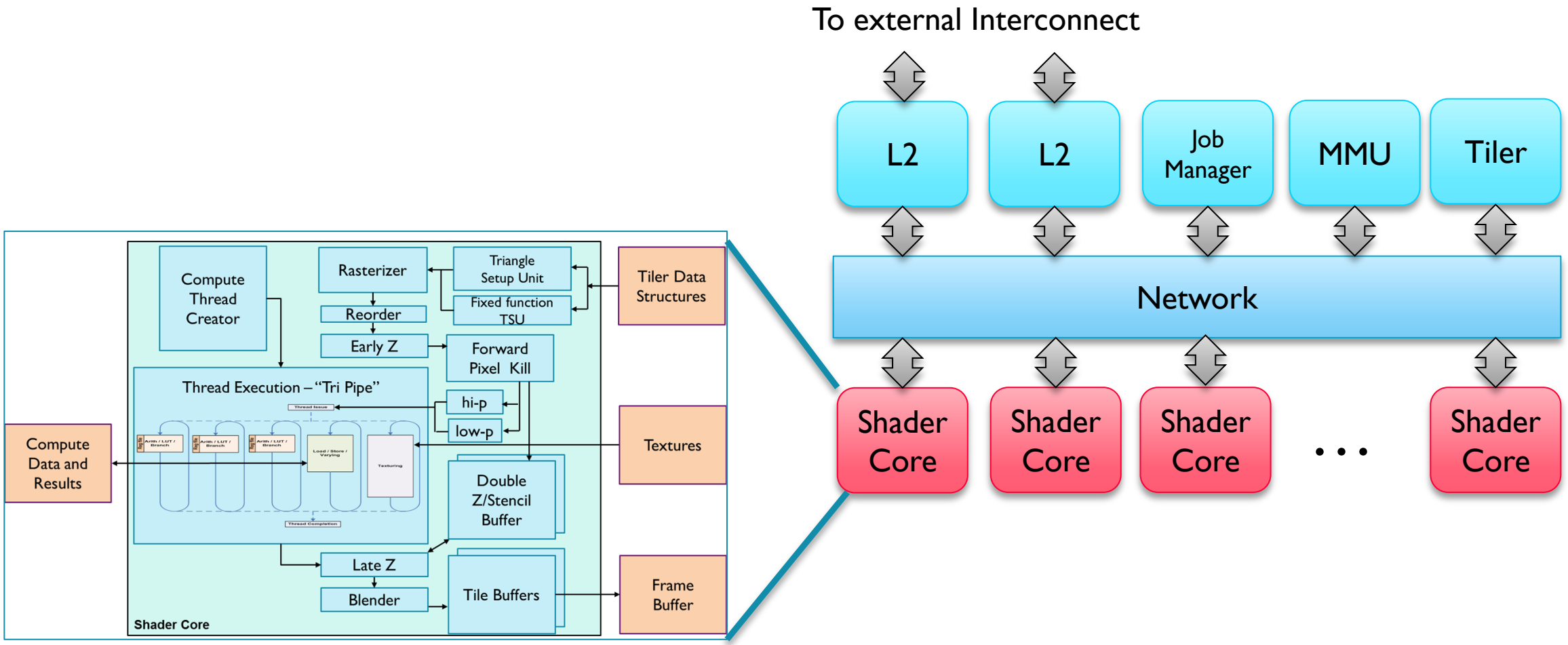
Mali-T880 GPU Block Diagram - Job Manager

- Fixed function block responsible for interfacing with the driver
  - Reads job descriptors from memory
  - Tracks inter-job dependencies
  - Distributes jobs across shader cores
  - Splits jobs into per-core tasks



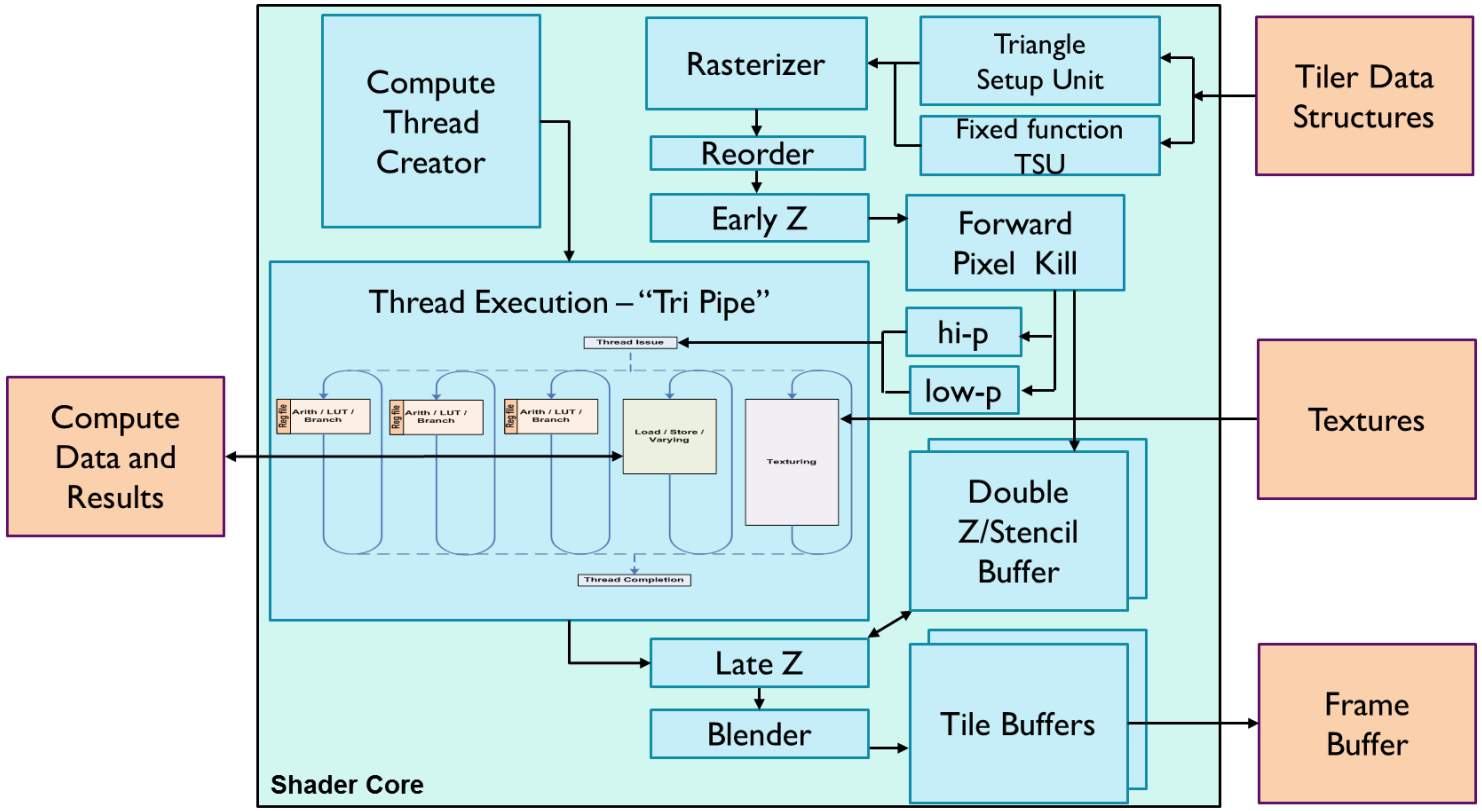
ARM®

Mali-T880 GPU Block Diagram - Shader Core



Mali-T880 GPU Block Diagram - Vertex Shading

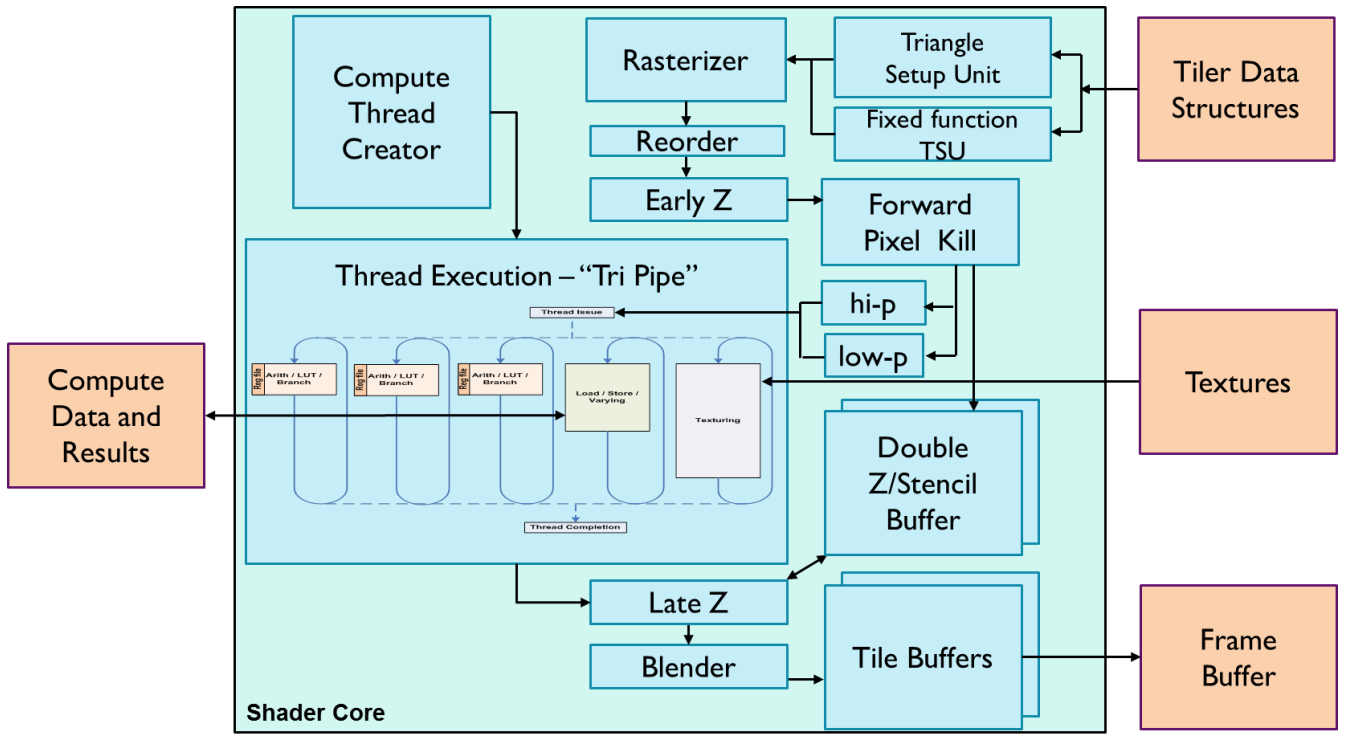
- Vertex shading threads are created by the compute thread creator
- Positions and attributes are read from memory
- Threads execute in the tri-pipe
- Data is written out to memory via an L1 cache
  - These L1 caches are coherent (across shader cores) and support atomics
- This same flow is used for compute kernels



ARM®

Mali-T880 GPU Block Diagram - Fragment Shading

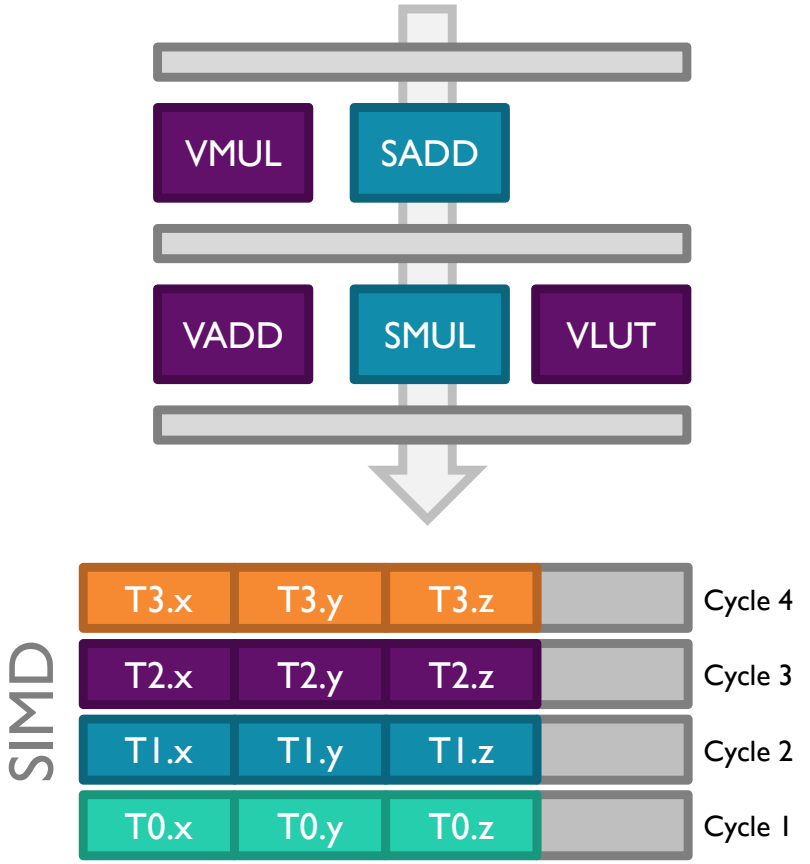
- The triangle setup unit reads the primitive lists generated by the tiler
- Primitives are sent to the rasterizer
- Z-tests are performed, and threads are created for each fragment
- Fragment threads execute in the tri-pipe
- Once complete, the final value or blend operation is sent to the Tile-Buffer to create the final color
- Once all primitives for a tile have been rendered, the tile-buffer writes the pixel values to memory.



ARM®

Mali-T880 GPU Shader Core - Arithmetic pipeline

- Arithmetic ISA on Midgard is SIMD + VLIW
  - Three vector units (128-bit datapath)
  - 4-lane FP32 or 8-lane FP16 for graphics
  - 16-lane int8 for compute
  - Two scalar units (32-bit datapath)
- One thread at a time executes in each pipeline stage



- Limited amount of out-of-order parallelism
  - Arith and Load/Store can progress under a pending Texture instruction

ARM®

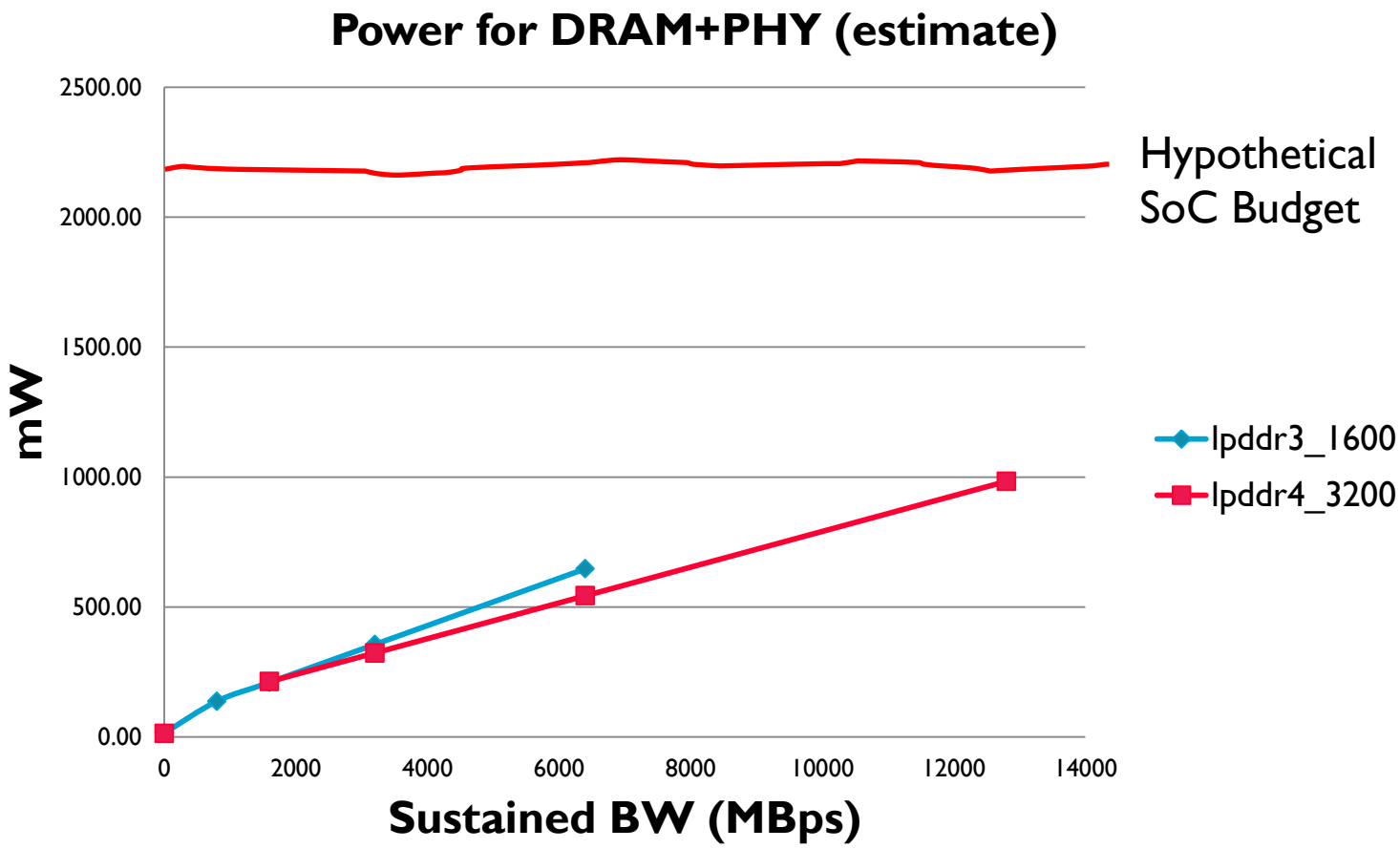
Mali-T880 GPU Shader Core - Power Management

- The Midgard hardware design supports two power management methods:
  - Clock gating
  - Power gating
    - The Job Manager provides a software controlled power manager interface for power gating
- Other power management facilities on the SoC can also be used
  - DVFS systems
    - Change the frequency and voltage of the GPU
  - Power controllers
    - Turn the entire GPU on or off

Mali-T880 GPU Shader Core - Power Gating

- Mali-T880 supports power gating of individual shader cores
- L2 cache can also be turned off once all cores have been turned off
- Controlled by software, but most work done by hardware
  - Shader core will only shutdown when current tasks have finished
  - L2 cache will only shutdown when they are clean
  - The faster you can turn off, the more power you'll save. Minimizes SW complexity.

Mali-T880 GPU Bandwidth Reduction



- Power used for DRAM/PHY is becoming a larger chunk of total SoC power.
- Higher performing GPUs use more BW
- In order to keep progressing GPU performance while also avoiding thermal limits, we need to find additional ways to save BW.

ARM®

Mali-T880 Bandwidth Savings - Transaction Elimination

- For some content, only a small portion of the scene changes from frame to frame

Typical GPU has to write out all tiles

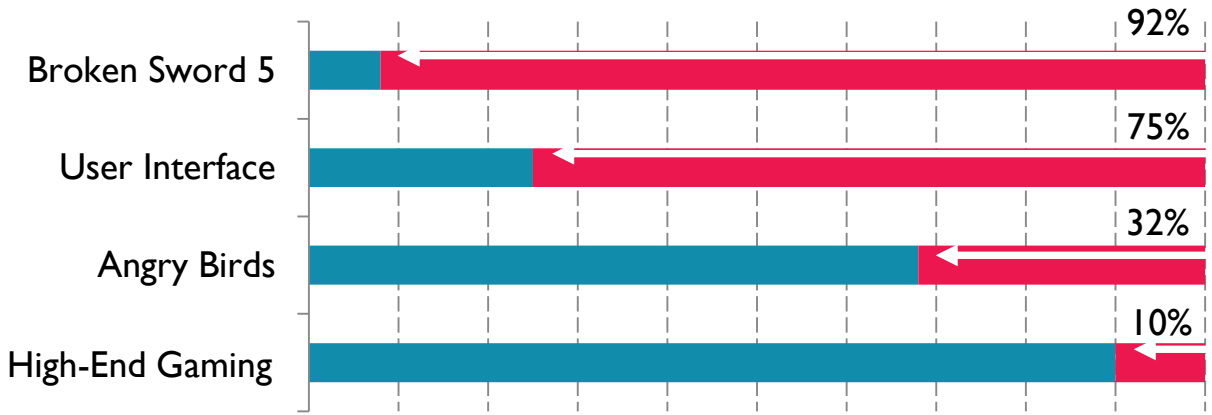
Green tile-overlay shows tiles are not changing

Maintain a list of signatures for each tile

Compare to sigs calculated for frame N+1

Where signatures match, don't write the tile

Output buffer savings with Transaction Elimination



Mali-T880 Bandwidth Savings - AFBC

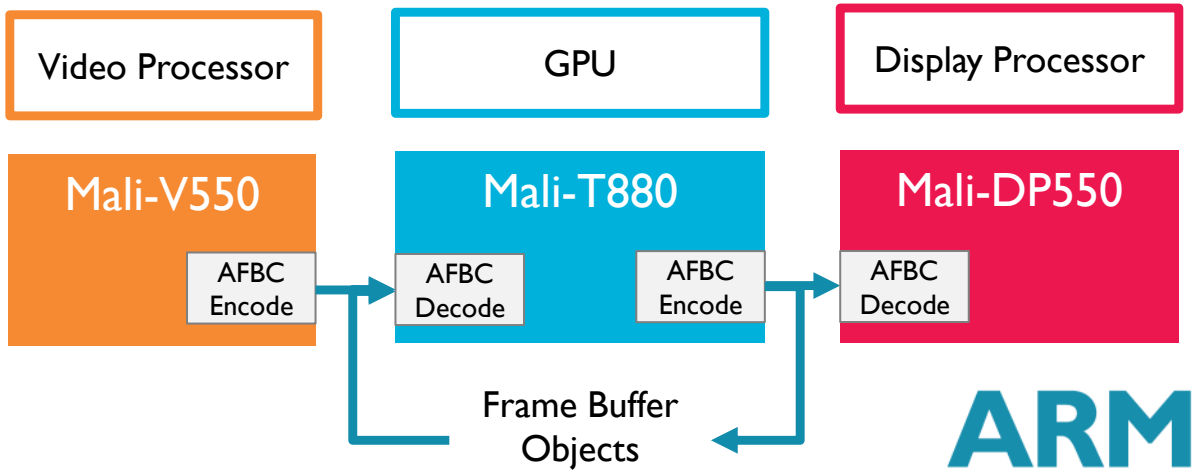
- Bandwidth reduction in Media System

- Reduced SoC energy consumption
  - Due to significant reduction in bandwidth
- Lossless compression format
  - Format preserves original image exactly (bit identical)

- Efficient implementations
  - Easy parallelization of encode and decode
- Supported in current and future Mali IP
  - GPU, Video and Display
  - Licensable for integration with 3rd Party media IP

System bandwidth reduction with AFBC

AFBC integration with media system



Mali-T880 Bandwidth Savings - ASTC

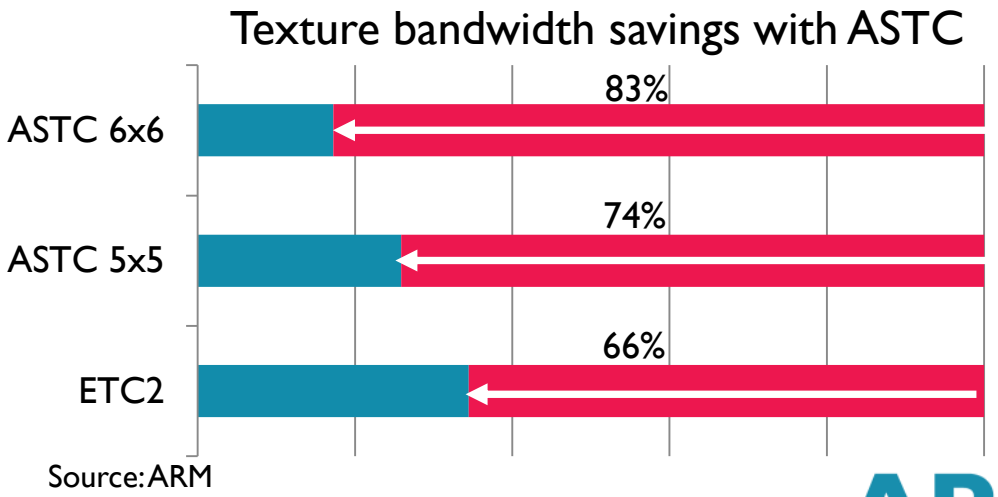
Flexibility, reduced size and improved quality

2D Codecs

High Dynamic Range

3D Textures

- ASTC is best-in-class compression
  - Scalable from 8 bits/pixel down to <1 bit/pixel
  - Orthogonal choice of formats (L, LA, RGB, RGBA)
  - 2D and 3D textures
  - LDR and HDR pixel formats, NPOT and MIPmap
  - Supports subimage upload on block boundaries
  - Bit exact decode



ARM®

Mali-T880 Conclusion

- Tile-based deferred rasterization with hierarchical tiling
- Pipelined rendering, overlapping vertex processing and tiling from one frame, with fragment processing from the previous frame
- Scalable from 1->16 shader cores, serving several markets.
- On-chip network enables ease of scalability
- Integrated MMU and fixed-function Tiler
- A pixel/cycle shader core serves as the fundamental building block, supporting simultaneous vertex and fragment shading.
- Designed from the ground up for power management
- Multiple BW saving techniques
  - Transaction Elimination
  - AFBC
  - ASTC